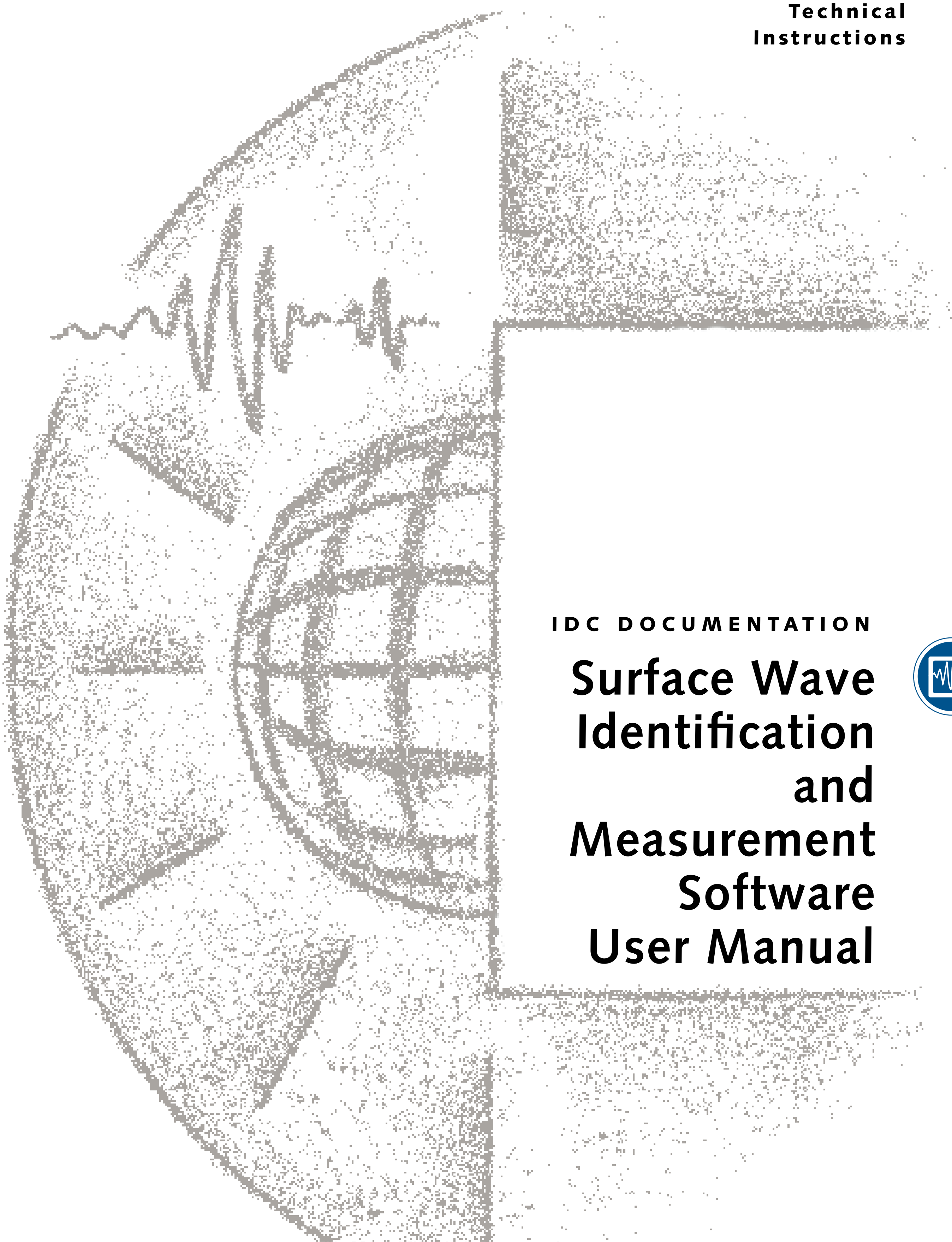
Technical
Instructions



IDC DOCUMENTATION

Surface Wave Identification and Measurement Software User Manual

Approved for public release;
distribution unlimited

Notice

Contributors

Jeffry L. Stevens, Science Applications International Corporation

Trademarks

Ordering Information

The ordering number for this document is SAIC-01/3014.

This document is cited within other IDC documents as [IDC6.5.16].

Surface Wave Identification and Measurement Software User Manual

CONTENTS

Surface Wave Identification and Measurement Software User Manual

FIGURES

Surface Wave Identification and Measurement Software User Manual

TABLES

About this Document

This chapter describes the organization and content of the document and includes the following topics:

- Purpose
- Scope
- Audience
- Related Information
- Using this Document

About this Document

PURPOSE

This document describes how to use the Surface Wave Identification and Measurement (SWIM) software of the International Data Centre (IDC). The software is part of the Post-location Processing computer software component (CSC) of the Automatic Processing Computer Software Configuration Item (CSCI) and is identified as follows:

Title:	Surface Wave Identification and Measurement
Abbreviation:	SWIM

SCOPE

This manual includes instructions for setting up the software, using its features, and basic troubleshooting. This document does not describe the software's design. This topic is described in [IDC7.1.3].

AUDIENCE

This document is intended for the first-time or occasional user of the software. However, more experienced users may find certain sections useful as a reference.

RELATED INFORMATION

The following documents complement this document:

- *Surface Wave Identification and Measurement* [IDC7.1.3]
- *Regionalized Maximum Likelihood Surface Wave Analysis* [Ste96]

- *Improved Methods for Regionalized Surface Wave Analysis* [Ste97]
- *Optimization of Surface Wave Identification and Measurement* [Ste01]

See "References" on page 67 for a list of documents that supplement this document. The following UNIX manual (man) pages apply to the existing SWIM software:

- *maxsurf*
- *LPcompile*

USING THIS DOCUMENT

This document is part of the overall documentation architecture for the IDC. It is part of the Technical Instructions category, which provides guidance for installing, operating, and maintaining the IDC systems. This document is organized as follows:

- Chapter 1: Introduction

  This chapter provides an overview of the software's capabilities, development, and operating environment.

- Chapter 2: Operational Procedures

  This chapter describes how to use the software and includes detailed procedures for startup and shutdown, basic and advanced features, security, and maintenance.

- Chapter 3: Troubleshooting

  This chapter describes how to identify and correct common problems related to the software.

- Chapter 4: Installation Procedures

  This chapter describes first how to prepare for installing the software, then how to install the executable files, configuration data files, and database elements. It also describes how to initiate operation and how to validate the installation.

- [References](#)

  This section lists the sources cited in this document.

- [Glossary](#)

  This section defines the terms, abbreviations, and acronyms used in this document.

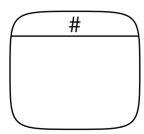
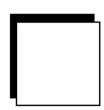
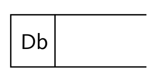
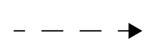
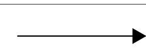
- [Index](#)

  This section lists topics and features provided in this document along with page numbers for reference.

Conventions

This document uses a variety of conventions, which are described in the following tables. Table I shows the conventions for data flow diagrams. Table II shows the conventions for entity-relationship diagrams. Table III lists typographical conventions.

TABLE I: DATA FLOW SYMBOLS

Description	Symbol[1]
process	
external source or sink of data	
data store Db = database store	
control flow	
data flow	

1. Symbols in this table are based on Gane-Sarson conventions [Gan79].

TABLE II: ENTITY-RELATIONSHIP SYMBOLS

Description	Symbol
One **A** maps to one **B**.	A ◄——► B
One **A** maps to zero or one **B.**	A ◄——○► B
One **A** maps to many **B**s.	A ◄——►► B
One **A** maps to zero or many **B**s.	A ◄——○►► B
database table	**tablename** *primary key* *foreign key* *attribute 1* *attribute 2* … *attribute n*

TABLE III: TYPOGRAPHICAL CONVENTIONS

Element	Font	Example
database table	**bold**	**arrival**
database table and attribute, when written in the dot notation		**msgdisc.***msgid*
database attributes	*italics*	*ampid*
processes, software units, and libraries		*maxsurf*
titles of documents		*Surface Wave Identification and Measurement*
computer code and output	`courier`	`Doing NBF test`
filenames, directories, and web sites		`MessageGet.par`
user-defined arguments and variables used in parameter (par) files or program command lines		`AUXDIR=`*message_path*`/w`
text that should be typed in exactly as shown		`maxsurf par=`*parfile*

Chapter 1: Introduction

This chapter provides a general description of the SWIM software and includes the following topics:

Chapter 1: Introduction

SOFTWARE OVERVIEW

Figure 1 shows the configuration hierarchy of the IDC software. The SWIM software is part of the Post-location Processing CSC in the Automatic Processing CSCI (which is "boxed" in the figure).

The SWIM software consists of a program, *LPcompile*, that prepares static input files (regionalized dispersion curves) for operations processing, program *maxsurf*, which is the main computational routine, and scripts (*MsInterval*, *MsOrid*, and *MsConflict*), which are used to prepare the inputs of and call *maxsurf* and *EvLoc*. *LPcompile* is run only when needed to update the dispersion curves. The scripts and *maxsurf* are run as needed in operations.

Figure 2 shows the relationship of the SWIM operations software to other components of the Automatic Processing CSCI and other CSCIs. Continuous data from the International Monitoring System (IMS) (external process "a" in Figure 2) are sent to the IDC where the Data Services CSCI formats the data and places them on disks with references to the data inserted into the operations database. The data are processed using the Station Processing and Network Processing CSCs of the Automatic Processing CSCI, which are controlled by the Distributed Processing CSCI. When the network processing has completed for a day of data, the results are reviewed by analysts using software from the Interactive Processing CSCI. Analysts may measure surface waves with the SWIM software for individual origins from the Analyst Review Station (*ARS*). After analyst review, the Post-location Processing CSC identifies and measures surface waves with the SWIM software, which uses the reviewed event locations as input. The results of the SWIM processing are placed in the operations database. Surface wave magnitudes are computed with *EvLoc*, which is not covered by this document (see [IDC7.1.5]).

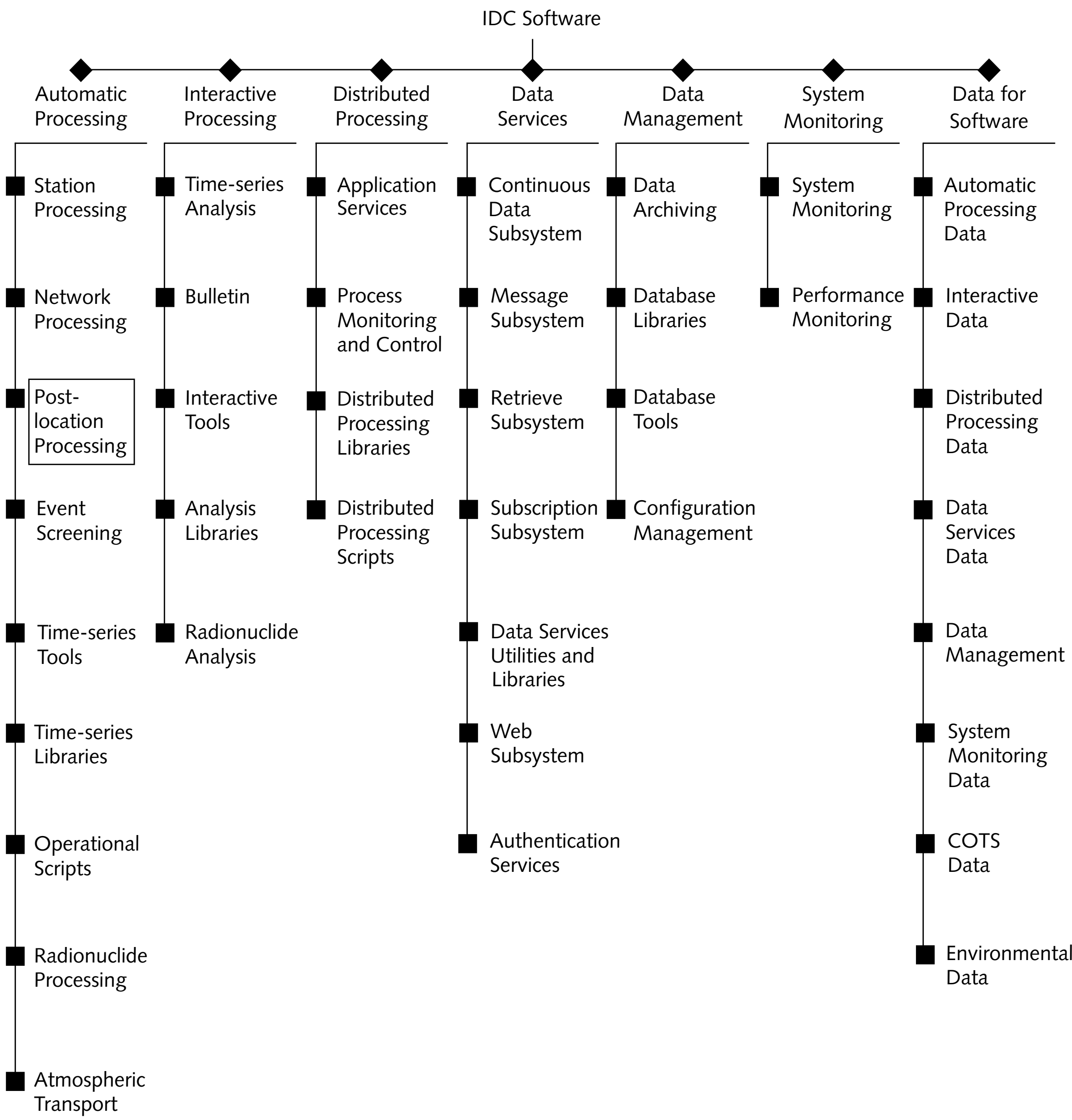


FIGURE 1. IDC SOFTWARE CONFIGURATION HIERARCHY

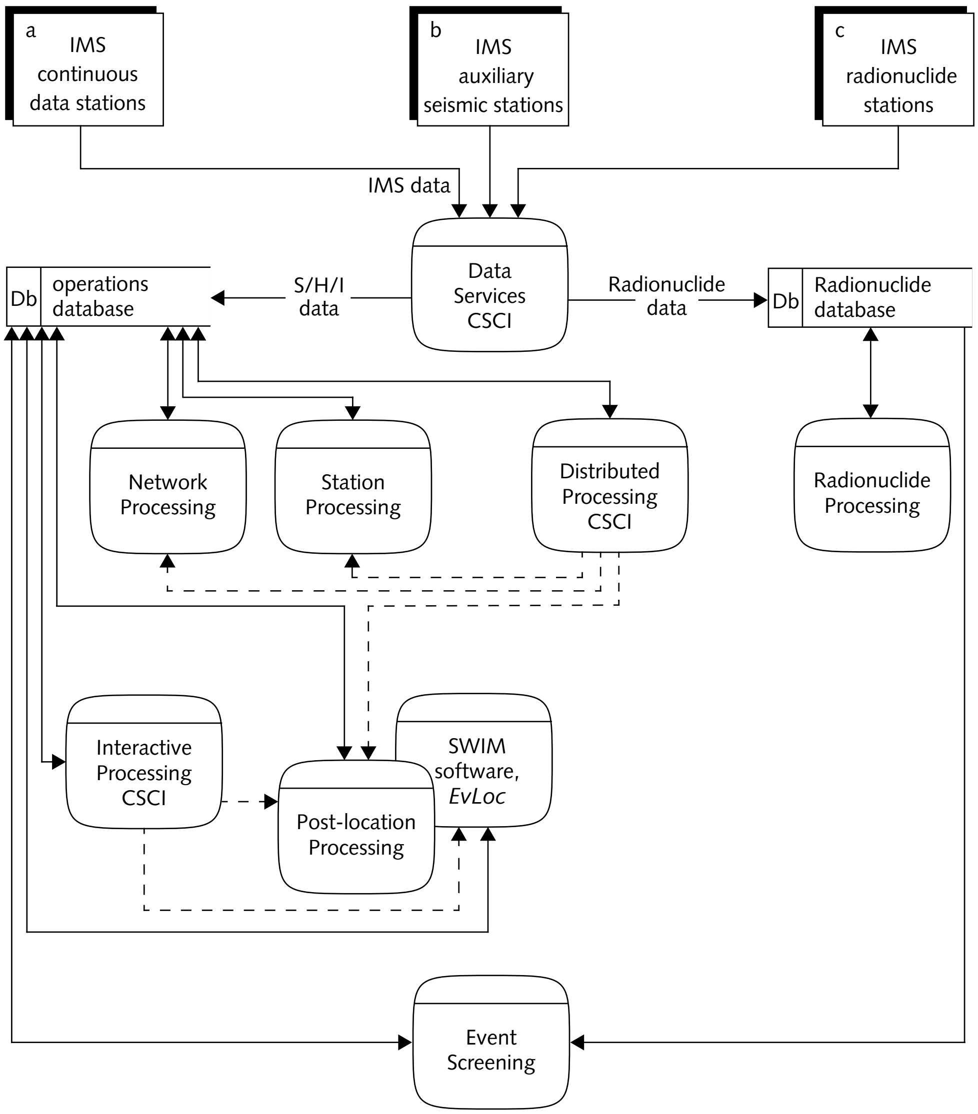


FIGURE 2. RELATIONSHIP OF SWIM SOFTWARE TO OTHER CSCIS AND AUTOMATIC PROCESSING CSCI SOFTWARE

STATUS OF DEVELOPMENT

The SWIM software was first implemented at the Prototype International Data Centre (PIDC) in May 1995. Several revisions have been made to the code and to the procedures used since that time. The software was included in the first delivery of software to the IDC in Vienna. Improvements in surface wave processing have been tested and implemented at the PIDC prior to installation in the IDC. The current version of the main SWIM software program (*maxsurf* 3.10) has been operational at the PIDC since 15 February 1997. Improved dispersion curves were installed in May 1998. The distance correction used to calculate M_s was changed to that of Rezapour and Pearce [Rez98] in September 1998 (*maxsurf* and *EvLoc* use the same formula).

FUNCTIONALITY

The SWIM software functions within the IDC processing system to identify and measure surface waves in all continuous long-period and broadband seismic data that are received by the IDC. The software includes a utility program, *LPcompile*, to convert regional dispersion curves to a binary format for use in automatic processing as well as the programs for processing the analyst-reviewed data.

The SWIM software is normally invoked in the automatic recall-processing pipeline; it may also be invoked by analysts during interactive processing. For recall processing the script *MsInterval* gets event origins to be processed in a given time interval. *MsInterval* then calls another script, *MsOrid*, for each origin within the interval. *MsOrid* gathers information about the origin from the database, uses depth and m_b constraints to determine if the origin should be processed, and passes the information to the main processing routine, *maxsurf*. During interactive processing, *MsOrid* can be called for individual origins directly from *ARS*.

The *maxsurf* program examines the arrival window where a surface wave would be expected (typically corresponding to a group velocity window of 5 km/s to 2 km/s) and applies a dispersion test to see if a surface wave can be identified. The dispersion test compares measured group arrival times in eight frequency bands to those predicted using a regionalized group velocity dispersion model. If the measured arrival times are consistent with the predicted arrival times, then an amplitude is measured and stored in the IDC database. Whether or not a surface wave is identified, *maxsurf* digitally replaces the instrument with a standard KS36000 long-period instrument and then searches for the largest amplitude in the 18–22 second time window within a time window determined from the predicted arrival time for this period range. This amplitude is then stored in the database identified as signal if a surface wave is found and as a noise measurement and upper bound on the surface wave amplitude otherwise. If no amplitude within this period range and time window can be measured, then no information is written to the database.

After *maxsurf* has completed its processing, *MsOrid* calls the *EvLoc* application, which estimates M_s and stores the results in the database (*EvLoc* is not covered by this document; see [IDC7.1.5]). Finally, another script, *MsConflict*, performs quality control of the results generated by *MsInterval* and *MsOrid*, removes spurious or misassociated arrivals from the database, and calls *EvLoc* to calculate the final M_s.

An example illustrates the processing of *maxsurf*. Figure 3 shows the location of an m_b 3.9 South Pacific earthquake that occurred on 15 June 1997, 12 IMS primary seismic stations within 100 degrees that recorded the event, and the great circle paths between the earthquake and the stations. Figure 4 shows the data recorded at these stations after conversion to a common (KS36000) instrument. The surface wave is visible at most of the stations; however, it is obscured by noise and difficult to see at some of them.

Figure 5 shows how surface waves are identified. A set of narrow band filters are applied to the data over a set of eight frequencies from 0.02 to 0.06 Hz. Long-period or broadband beams are formed at arrays using the known azimuth and approximate slowness. The arrival times at each frequency are compared with predicted arrival times generated from the regionalized group velocity model. Figure 5 shows the bounds on the allowed dispersion (lines) and the measured group velocities (plus signs) at the eight frequencies for several stations. The dispersion test requires that six of the eight measured data points lie within the predicted bounds. All stations except TXAR meet this requirement.

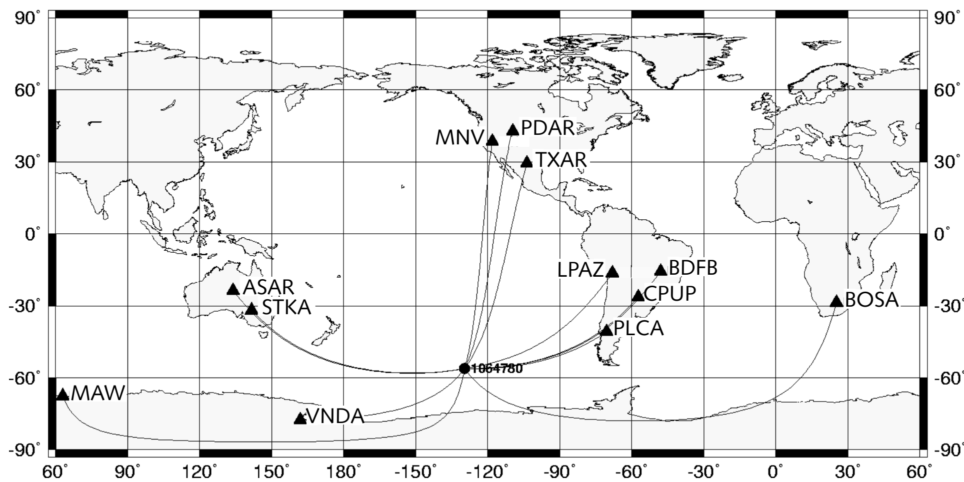


FIGURE 3. 1997 SOUTH PACIFIC EARTHQUAKE AND IMS STATIONS

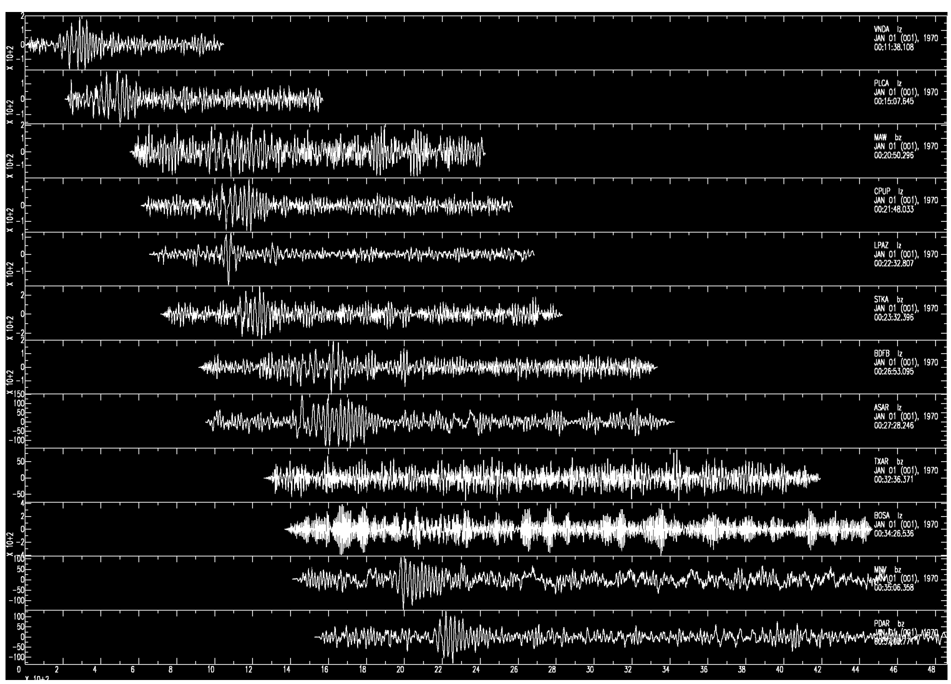


FIGURE 4. LONG-PERIOD DATA FROM 1997 SOUTH PACIFIC EARTHQUAKE

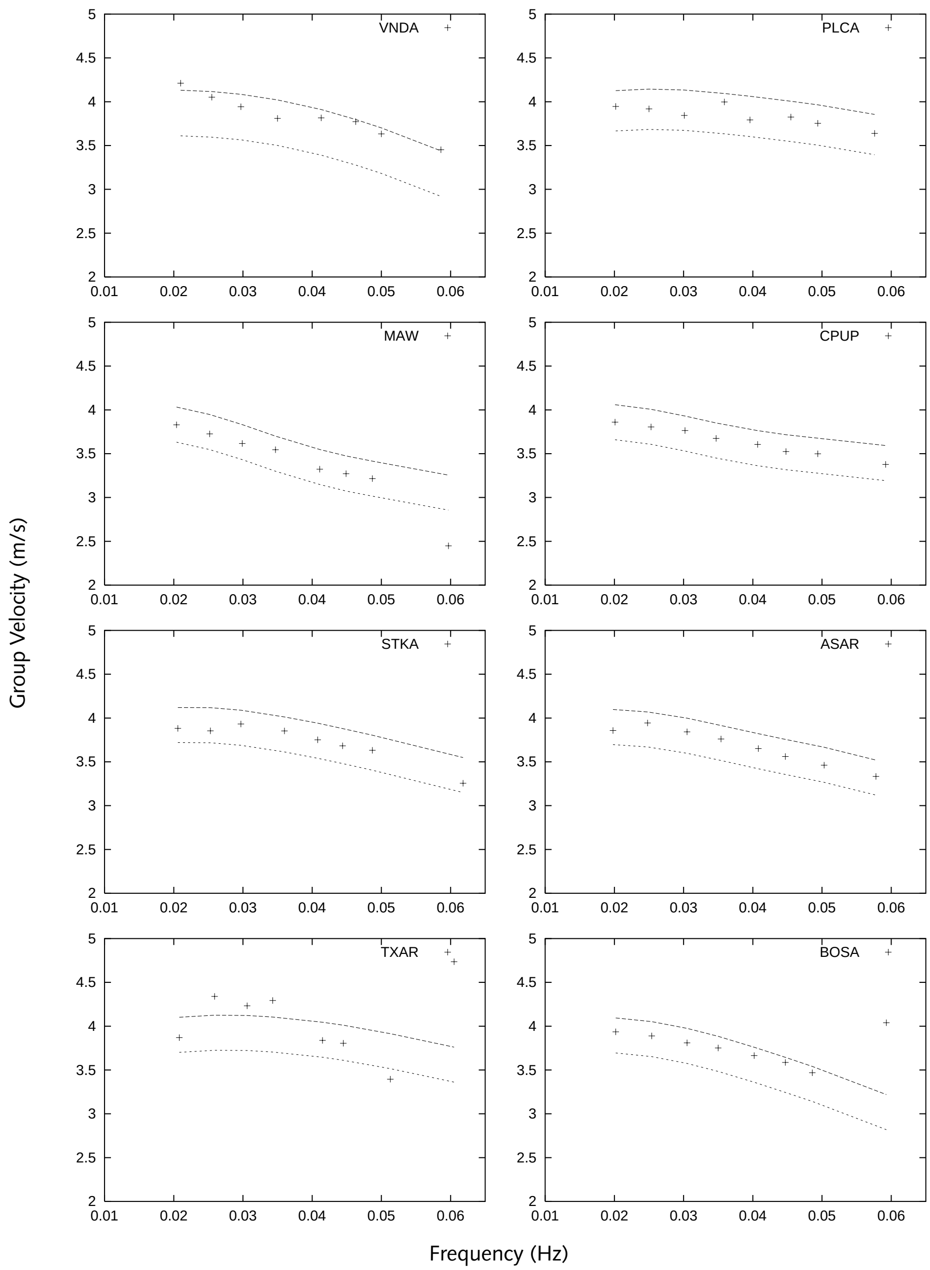


FIGURE 5. GROUP VELOCITY MEASUREMENTS AND PREDICTED DISPERSION WINDOW OF 1997 SOUTH PACIFIC EARTHQUAKE

Features and Capabilities

The SWIM software is designed to process long-period and broadband data automatically, mostly under the control of the Distributed Application Control System (DACS) (see [IDC6.5.2Rev0.1]). This feature ensures that little, if any, user intervention is necessary. User intervention is required only for quality control, for starting and stopping the automatic processing stream, and for execution of the *LPcompile* and *MsConflict* scripts.

The SWIM software obtains input data and writes output to the operations database tables of the IDC schema [IDC5.1.1Rev2]. The SWIM software processing is based on the origin identifier (*orid*). A list of *orid*s is extracted for the processing interval and is then provided as input for surface wave processing. Event, station, waveform, and instrument response information is extracted from the database using the *orid*. The processing output is written to the database using the *orid*.

The SWIM software uses path-dependent dispersion to predict the observed group arrivals. As models improve, so does the ability to predict the dispersion.

Performance Characteristics

Typical processing time for the main SWIM processing program, *maxsurf*, is a few (~4) seconds per station per event on a Sun UltraSparc computer. On a typical day, approximately 50 events are identified. Assuming that each event is recorded by 20 long-period or broadband stations within 100 degrees of the event's location, approximately 1,000 station/event pairs are processed per day. Therefore, processing time for a typical day of data is expected to be approximately two hours. However, other factors could increase this time significantly. The SWIM software performs a number of database queries, so time spent waiting for database response adds to the total processing time. The main SWIM processing program, *maxsurf*, requires only one database connection for each event to minimize the effect of database response time. File input/output (I/O) can be another potential bottleneck because *maxsurf* must read a large number of waveforms. If data are stored on a local disk, this is not a significant problem; but if data are obtained from an archive data storage system, the total processing time can increase by an order of magnitude.

Related Tools

The SWIM software identifies and measures surface waves, but another program, *EvLoc* [IDC7.1.5], is used to compute the surface wave magnitude that is inserted into the database and that appears in IDC bulletin products.

INVENTORY

Table 1 lists the items needed to operate the SWIM software. Parameter files and scripts for configuring the DACS and *ARS* are not included.

TABLE 1: SWIM SOFTWARE INVENTORY

Item	Type
LPcompile	executable
`LP_grid.LR`	Rayleigh wave (LR) grid description file (input to *LPcompile*)
`LP_vel.LR`	LR dispersion file (input to *LPcompile*)
MsInterval	script
MsOrid	script
MsConflict	script
`MS.par`	parameter file for *MsInterval*, *MsOrid*, and *MsConflict*
maxsurf	executable
`maxsurf.par`	parameter file for *maxsurf*
libLP	library
liblparray	library
liblppar	library
libmag	library

TABLE 1: SWIM SOFTWARE INVENTORY (CONTINUED)

Item	Type
libnbf	library
libpathcor	library
liblpmotion	library
libresp	library
libs3obj	library
libutilc	library
libutilf	library
affiliation	database table
amplitude	database table
arrival	database table
assoc	database table
instrument	database table
netmag	database table
origerr	database table
origin	database table
parrival	database table
sensor	database table
site	database table
sitechan	database table
stamag	database table
wfdisc	database table

ENVIRONMENT AND STATES OF OPERATION

The SWIM software is designed to run in a UNIX environment either as part of an automatic system or as individual processing modules and scripts. Most of the programs and scripts have a command line interface that accepts parameter files in the standard format recognized by *libpar*.

Software Environment

The SWIM software is designed to run on a UNIX workstation such as the Sun UltraSPARC 5. Typically, the hardware is configured with 128 MB of memory and a minimum of 4 GB of magnetic disk. The compiled software modules, *maxsurf* and *LPcompile*, and associated scripts require approximately 1 MB of disk space. Regionalized dispersion curves require approximately 5 MB of disk space. The SWIM software writes diagnostic information to standard output. If the information is saved, the disk space used scales with the number of stations in the network and the number of events processed (approximately 50 KB per event are produced with the current IMS network). The SWIM software is designed for the Solaris 7.0 operating system and ORACLE 8.1.5 running on Sun workstations and servers. Distributed processing is controlled using BEA TUXEDO software.

Normal Operational State

Except for *LPcompile* and *MsConflict*, the SWIM software runs without user intervention. The SWIM software's interface with other IDC systems is through the database, the DACS, and startup calls in the Interactive Processing CSCI.

When analysts allocate time, an interval is inserted into the **interval** database table for the allocated time. As different stages of the review are completed, **interval**.*status* is updated. The DACS starts different processes depending on the value of **interval**.*status.* A *tuxshell* calls *MsInterval*, which begins the surface wave processing as part of recall processing.

Analysts may also invoke *MsOrid* for individual origins through *ARS*. A button in *ARS* sends a message to a *tuxshell*, which calls *MsOrid*.

MsConflict is not managed by the DACS, but is started manually by the senior analyst just prior to preparing the REB.

Contingencies/Alternate States of Operation

The main SWIM software processing program, *maxsurf*, can be run independently of the scripts used to invoke the program in the operational system. This alternative is often useful for testing, for optimizing parameters, or for checking results for individual events of interest.

Chapter 2: Operational Procedures

This chapter provides instructions for using the software and includes the following topics:

- Software Startup
- Software Shutdown
- Basic Procedures
- Advanced Procedures
- Maintenance
- Security

Chapter 2: Operational Procedures

SOFTWARE STARTUP

All SWIM software can be started from a command line. In normal operations, use the command line directly only for *LPcompile* and *MsConflict*. The other programs are started through the DACS. The DACS starts *MsInterval*, which starts *MsOrid* for each *orid*. *MsOrid* starts *maxsurf* if processing is required. Analysts may also start *MsOrid* directly from *ARS* through the DACS.

LPcompile

To execute *LPcompile*, use the following command:

`LPcompile` *LPdir gridfile velfile* [*compfile*]

LPdir is the directory containing the ASCII format dispersion curves. *gridfile* is the dispersion file that describes the grid and associates a particular dispersion model with a particular location on the grid. *velfile* is the dispersion file that contains all of the frequency-dependent dispersion curves. *compfile* is the name of the output binary file that contains the information in both *gridfile* and *velfile*. If *compfile* is omitted, then the output binary file is named *velfile*`.o`, where *velfile* is the input dispersion filename.

MsInterval

MsInterval uses the *getpar.pl* Perl library to provide a standard par interface. To execute *MsInterval* from the command line, use the following command:

`MsInterval par=`*parfile*

From *tuxshell*, *MsInterval* is called with the following command:

MsInterval par=Ms.par start-time=*epochstart* end-time=*epochend*

MsOrid

MsOrid uses the *getpar.pl* Perl library to provide a standard par interface. To execute *MsOrid* from the command line, use the following command:

MsOrid par=*parfile*

maxsurf

To execute *maxsurf* from the command line, use the following command:

maxsurf par=*parfile*

parfile is the standard format parameter file containing all *maxsurf* input parameters. When *maxsurf* is started by *MsOrid*, the *parfile* contains only the information that is static for each event. The origin identifier (ID), station identifiers for long-period or broadband channels, and database account are passed as command line arguments as follows:

maxsurf par=*parfile* net_list=*nlist* orid=*orid* chan_list=*clist* \
in_db=*acct*

MsConflict

MsConflict uses the *getpar.pl* Perl library to provide a standard par interface. To execute *MsConflict* from the command line, use the following command:

MsConflict par=*parfile*

SOFTWARE SHUTDOWN

The SWIM software normally terminates after processing all events for a time interval or for the specific event being processed. No action is required. If necessary, use the DACS to obtain an orderly shutdown of the SWIM software (see [IDC6.5.2Rev0.1]).

If you terminate the main SWIM processing program, *maxsurf*, manually using a system `kill` command, partially processed information may be left in the database. These data are removed by *maxsurf* when it is restarted, prior to writing new results for the event.

BASIC PROCEDURES

Most SWIM software run automatically under the control of the DACS and have no basic procedures. The basic procedures for *LPcompile* and *MsConflict*, which are run independently, are the procedures for starting the software and the schedule for running the software. Run *LPcompile* any time that the regionalized dispersion curves are changed and any time that the grid file is changed. *MsConflict* produces the final surface wave results and must be run before any bulletin that includes surface wave magnitudes is released.

Obtaining Help

To obtain help for using the SWIM programs and parameters refer to the man pages for the components of the subsystem, [IDC7.1.3] and [IDC6.5.2Rev0.1].

ADVANCED PROCEDURES

This section provides detailed instructions for using the software's advanced features.

Adding/Deleting Stations

Any station whose data are processed by the SWIM software must have station information in the appropriate database tables and must have accurate instrument response information. When a new station is added to the IDC operations, the station information is inserted into the database. Check that the station information in the database is correct, that the corresponding instrument response file is installed, and that the response information is accurate.

The SWIM par files must be properly configured when adding a new station. Make the following par file changes:

1. If the new station data should be limited to the bz channel, add the name of the new station (single quoted and comma separated) to *bzonly_list* in the *MsInterval* par file (`Ms.par`).
2. The *net_list* parameter for *maxsurf* in `maxsurf.par` normally is set by *MsOrid* using database queries; no *maxsurf* parameter changes are necessary. However, if the call to *maxsurf* in *MsOrid* includes a list of stations, then add the new station to the list.

Optimizing Parameters

Narrow-band filtering and the dispersion test require that a few processing parameters be defined and optimized. The following list describes these parameters and their optimum values as determined by processing a large body of GSETT-3 data and reviewing the results.

- Narrow-band filtering frequencies are used on the data to compare with predicted group velocity arrival times.

  The following eight frequencies are good average global values: 0.02, 0.025, 0.03, 0.035, 0.04, 0.045, 0.05, 0.06 Hz. These frequencies could be adjusted and performance possibly improved by using higher frequencies at shorter distances. In Figure 5 on page 8 the measured frequency is slightly different from the input frequency. This occurs because the finite width of the narrow-band filter combined with amplitude variation of the spectra causes a shift in the instantaneous frequency of the output time series.

- The allowable error in the group velocity arrivals and the fraction of group velocities that are required to match the predicted arrival times define the criteria for declaring a surface wave detection.

  A minimum of 70% of the group velocity points, six out of eight in this case, must be within the predicted arrival time window. This is sufficient to remove most "accidental" arrivals where the arrival peaks of noise just happen to be within the arrival window. Frequently, one or two arrival times are out of range because of low signal-to-noise ratio (snr), interference, or other factors (see Figure 5 on page 8, for example), and this requirement allows such arrivals to be identified. The group arrival time *t* must be within the time window given by equation (1):

$$\frac{r}{(v_p + v_0)} - p_0 T - t_0 < t < \frac{r}{(v_p + v_0)} + p_0 T + t_0 \tag{1}$$

  where r is the source to receiver distance, v_p is the predicted group arrival time, T is the period, and v_0, p_0, and t_0 are user-definable constants. The parameters currently used are $v_0 = 0.2$, $p_0 = 1.0$, and $t_0 = 0$. This has the effect of changing the allowed group velocity window from about 0.2 km/s at large distances to about 0.3 km/s at regional distances. A smaller group velocity window at large distances is necessary because a fixed group velocity window corresponds to a large time window at large distances, increasing the chance for spurious arrivals. Along with the group arrival, the surface wave amplitude must be measured within the time limits defined by equation (1).

- The narrow-band filter Q determines the characteristics of the narrow-band filters.

  The narrow-band filter F(f) is defined by equation (2):

$$F(f) = e^{-\alpha(f - f_c)^2} \tag{2}$$

  where

$$\alpha = \frac{\ln 2}{2}\left(\frac{Q}{f_c}\right)^2 \quad (3)$$

and f_c is the center frequency. In general, Q should be smaller for close distances and larger for large distances, with a reasonable range being from about 8 to 20. A lower Q value gives better time resolution, and a higher Q smoothes the time series for more distant seismograms that have traveled along complex paths. A narrow-band filter Q of 15 (the default) is a good average value to use over a wide distance range.

These parameters are specified in the parameter file in the operational system and you may adjust (or tune) them to improve detection. For example, identification at regional distances may be improved by using higher frequencies for the dispersion test and a narrower time window at very short distances.

MAINTENANCE

You may optionally archive or delete log files, which are written to standard output. The SWIM software does not create temporary database tables, so database maintenance is not necessary.

SECURITY

Maxsurf requires the ORACLE user identification (ID) and password to establish a connection to the ORACLE database. Passwords are created and modified by the ORACLE system administrator.

Chapter 3: Troubleshooting

This chapter describes how to identify and correct problems related to the SWIM software and includes the following topics:

- Monitoring
- Interpreting Error Messages
- Solving Common Problems
- Reporting Problems

Chapter 3: Troubleshooting

MONITORING

The SWIM software log files can be useful for examining processing details and for diagnosing problems. The excerpt following this paragraph is from a typical *maxsurf* log file. The log shows processing of two stations, BDFB and BGCA, for an event with origin ID 20318563. For both stations, the log file contains distance and azimuth information. Station BDFB is out of range, so the station is not processed further. For BGCA, the processing continues and *maxsurf* performs narrow-band filtering, three-component (3-C) polarization filtering, and magnitude reasonableness tests on the data. The waveform start time, sampling rate, and number of points within the selected time interval (based on a group velocity window of 5 km/s to 2 km/s) are given, followed by the predicted and observed dispersion curves. The columns for the dispersion (NBF) test are input frequency, instantaneous (measured) frequency, amplitude, arrival time relative to the waveform start time, arrival time relative to origin time, measured group velocity, and predicted group velocity. In this case, seven of the eight group velocities are within the required bounds, which is shown as a percentage on the "`OKfrac`" line (88 percent of the measurements are within bounds). The "`Dispersion SW Test:1`" line indicates that the waveform passed the dispersion test. The estimated back azimuth is 354 degrees, compared to a known back azimuth of 4 degrees, so the estimated back azimuth is within 11 degrees of the true back azimuth. However, the azimuth test is not used as an existence test. The line "`Magnitude SW test: 1`" indicates that the magnitude of 4.34 was above the minimum permissible magnitude of 1.0, and the line "`MAGB SW test: 1`" indicates that the magnitude of 4.34 was reasonable, based on historical values, for an event with m_b 4.44. The arrival would have been rejected for an event with m_b less than 3.53 at this distance (83 degrees). An arrival with M_s greater than 5.94 would also be rejected for an event with m_b 4.44.

```
Net BDFB, Stations BDFB
orid 20318563, otime  917844761.2, olat    85.51, olon    86.45, depth    0.0, mb  4.44
sta BDFB, slat    -15.64, slon    -48.01
distance   12084.64, delta    108.68, azimuth    313.46, bazimuth      3.40
Station BDFB skipped, out of range

Net BGCA, Stations BGCA
orid 20318563, otime  917844761.2, olat    85.51, olon    86.45, depth    0.0, mb  4.44
sta BGCA, slat      5.18, slon     18.42
distance    9249.61, delta     83.18, azimuth    248.46, bazimuth      4.22

Read data: 2776 points, dt    1.000, start time  917846611.4

Doing NBF test
     fcntr  inst_freq        amp     tpkrel     tpkabs     grpvel  predicted
    0.0200     0.0203  7.368e+01      586.5     2436.8      3.796      3.778
    0.0250     0.0257  1.124e+02      682.8     2533.1      3.652      3.623
    0.0300     0.0299  1.579e+02      793.1     2643.4      3.499      3.481
    0.0350     0.0331  8.491e+01      882.5     2732.8      3.385      3.370
    0.0400     0.0419  8.280e+01     1213.6     3063.9      3.019      3.113
    0.0450     0.0445  1.209e+02     1226.5     3076.7      3.006      3.053
    0.0500     0.0490  8.931e+01     1281.9     3132.2      2.953      2.966
    0.0600     0.0584  4.254e+01     1725.4     3575.7      2.587      2.830

Dispersion test: OKfrac 0.88; GVdiff -0.04, Sdev 0.09; Tdiff 50.3, Sdev 110.8
Dispersion SW test: 1

Doing Polarization test
fmin      0.020, fmax      0.05, imin 82, imax 205
fstat    9.716, azim   353.64, azres   -10.57, rmsAmp  1.961e+05, ellip    0.827

Calculating magnitude
mag     4.34, amp   1.75e+02, period    19.51, arrtime    3141.13, pred_time    3157.50
Magnitude SW test: 1, mag      4.34
MAGB SW test: 1, mb   4.44, mbmin   3.53, ms   4.34, msmax   5.94

Surface Wave score 3/3: detected 20318563 BGCA

4 records committed to database
```

INTERPRETING ERROR MESSAGES

Message:	`Dropped array channel 0, amp 0.000000e+00, median 2.677034e+08`
Description:	*maxsurf* checks the amplitude of all signals before generating a beam and drops any signals that are inconsistent with the median amplitude. This error message indicates that a channel has been dropped.
Action:	None required, the program is operating normally for processing of array data with a dead or malfunctioning channel.
Message:	`WSDgetData Error: no data read. get_wfdisc_tuples: No wfdisc tuples found.`
Description:	*maxsurf* tried to read data from the **wfdisc** file and either could not find an entry for the station in the **wfdisc** table or could not read the data file pointed to by the **wfdisc** entry.
Action:	The error indicates missing data. No action is required unless new data are added to the system at a later date, in which case the event should be reprocessed.
Message:	`fill_wfm: WARNING: Samprates different: first_gp_wfd_rate=1.000000; wfmap[0].wfd[2]->samprate=1.000000.`
Description:	*maxsurf* often must piece together waveforms from multiple **wfdisc** entries. Sometimes they are at different sampling rates.
Action:	Usually the sampling rate difference only indicates roundoff error, in which case no action is required. *maxsurf* does not operate correctly if the sampling rates differ significantly.

SOLVING COMMON PROBLEMS

The SWIM software rarely, if ever, terminates abnormally. However, if the system crashes, or if the database stops responding during processing, *maxsurf* may leave partial results in the database. In this case it is necessary to restart *maxsurf*. Earlier results are deleted by *maxsurf* prior to storing new results in the database.

REPORTING PROBLEMS

The following procedures are recommended for reporting problems with the application software:

1. Diagnose the problem as far as possible.
2. Record information regarding symptoms and conditions at the time of the software failure.
3. Retain copies of the relevant sections of the application log files.
4. Contact the provider or maintainer of the software for problem resolution if local changes of the environment or configuration are not sufficient.

Chapter 4: Installation Procedures

This chapter provides instructions for installing the software and includes the following topics:

Chapter 4: Installation Procedures

PREPARATION

Prior to installation, the ORACLE database with the standard IDC tables should be installed and available over the network. The machine that runs the SWIM software should have Solaris 7.0 or later installed, and it must be connected to the network in a manner that allows read and write queries to be performed on the ORACLE database. Access to ORACLE through Perl scripts must also be available.

Obtaining Released Software

Obtain the software via FTP from a remote site or via a physical medium, such as tape or CD-ROM. The software and associated configuration data files are stored as one or more tar files. Transfer the tar files to an appropriate location on a local hard disk and untar the files into a standard UNIX directory structure.

Mapping Hardware

Select the hardware on which to run the software components. Software components are generally mapped to hardware to be roughly consistent with the software configuration model.

EXECUTABLE FILES

Install the two executable files, *maxsurf* and *LPcompile*, and Perl scripts *MsInterval*, *MsOrid*, and *MsConflict* in a location accessible to the operational system.

CONFIGURATION DATA FILES

Three types of configuration files must be installed: instrument response files for all stations to be processed, regional dispersion curve files, and par files. You must configure the *maxsurf* parameter file to correspond to the filesystem paths of the dispersion curve and instrument response files.

LPcompile

LPcompile uses two input files: *gridfile* is a dispersion file that describes the grid and associates a particular dispersion model with a particular location on the grid, and *velfile* is the dispersion file that contains all of the frequency dependent dispersion curves. The *gridfile* contains the number of latitude and longitude samples and the spacing between geographic points at the top. This is followed by the indexes of the *velfile* dispersion model assigned to each grid point. The indexes are grouped by co-latitude range and are listed in increasing longitude range order. An example of a *gridfile* follows:

```
# #Lats  #Lons  Spacing (deg.) for grid indices
    36     72      5.0

# Co-Latitude Range:   0.00 to 5.00 deg.
150 150 150 150 150 150 150 150 150 150 150 150 150 150 1
50 150 150 150 150 150 151 151 151 151 151 151 151 150 15
0 150 150 150 151 151 151 151 151 151 151 151 151 129 129
 129 129 129 151 151 151 151 151 151 151 151 151 151 151
151 151 151 151 151 151 151 151 151 151 151 151 151 151 1
51

# Co-Latitude Range:   5.00 to 10.00 deg.
150 150 151 151 151 151 151 151 113 113 113 113 113 113 1
51 151 151 151 151 151 151 151 150 150 150 151 151 151 15
1 151 151 151 139 139 139 151 151 151 151 151 151 151 151
 151 151 139 139 139 139 139 139 139 139 139 124 43 43 43
 43 43 44 44 44 44 44 45 44 44 43 23 151 151
```

```
# Co-Latitude Range:  10.00 to 15.00 deg.
2 21 118 118 23 24 24 25 25 26 26 25 25 25 25 25 25 24 23
 118 118 118 23 112 113 113 112 23 23 23 23 112 112 113 1
39 139 151 151 23 24 139 139 139 139 139 139 139 115 122
122 122 122 122 124 43 43 118 24 43 44 47 47 48 48 47 46
45 44 25 26 24 3
```

… (one group for each co-latitude range up to 85.0 to 90.0 degrees)

The *velfile* lists the number of model index samples and the number of period samples. This is followed by the periods (in seconds) that the period samples represent (for example, if there are 100 period samples, then 100 periods are listed). The remainder of the file consists of the models (one for each index) with the phase velocity at each of the period samples. An example of a *velfile* follows:

```
# LR parameter table discretized as index/period
 154   # number of index samples
 100   # number of period samples
5.000  5.051  5.102  5.155  5.208  5.263  5.319  5.376
5.435  5.495  5.556  5.618  5.682  5.747  5.814  5.882
5.952  6.024  6.098  6.173  6.250  6.329  6.410  6.494
6.579  6.667  6.757  6.849  6.944  7.042  7.143  7.246
7.353  7.463  7.576  7.692  7.812  7.937  8.065  8.197
8.333  8.475  8.621  8.772  8.929  9.091  9.259  9.434
9.615  9.804  10.000  10.204  10.417  10.638  10.870
11.111  11.364  11.628  11.905  12.195 1 2.500  12.821
13.158  13.514  13.889  14.286  14.706  15.152  15.625
16.129  16.667  17.241  17.857  18.519  19.231  20.000
20.833  21.739  22.727  23.810  25.000  26.316  27.778
29.412  31.250  33.333  35.714  38.462  41.667  45.455
50.000  55.556  62.500  71.429  83.333  100.000  125.000
166.667  250.000  500.000
# A0    normal oceanic, 0.15 km seds.
1.285697
```

```
1.282573
1.279342
1.276001
1.272544
1.268967
1.265263
1.261428
1.257454
1.253336
```
… (one sample for each period)

```
# A1    normal oceanic 0.5 km seds.
1.202370
1.200918
1.199333
1.197610
1.195747
1.193738
1.191579
1.189264
1.186789
1.184148
```
… (one sample for each period, one group for each model index)

MsInterval, MsOrid, and MsConflict

Most of the parameters for the *MsOrid*, *MsConflict*, and *MsInterval* scripts are common; therefore, you must maintain only one par file. Because the par file is shared by three separate scripts, no defaults are supported to ensure that defaults do not differ between scripts. The parameters in the *MsOrid*, *MsConflict*, and *MsInterval* par file follow:

I/O Parameters

I/O parameters define the data to process and where the results are written.

start-time

Epoch start time of data interval.

end-time

Epoch end time of data interval.

account

Database account.

out_arrival_table

Output **arrival** table.

out_assoc_table

Output **assoc** table.

out_netmag_table

Output **netmag** table.

out_origerr_table

Output **origerr** table.

out_origin_table

Output **origin** table.

out_parrival_table

Output **parrival** table.

out_stamag_table

Output **stamag** table.

out_amplitude_table

Output **amplitude** table.

Control Parameters

Control parameters define how to process the data.

magtype

Magnitude type written to the **netmag** table (single quoted).

phaselist

Valid phase types in the **assoc**, **parrival**, and **stamag** tables (single quoted, comma separated).

amptype

Valid amplitude types in the **amplitude** table (single quoted, comma separated).

excludelist

Stations excluded from processing (single quoted, comma separated).

bzonly_list

Stations for which the bz channel is used (single quoted, comma separated).

overwrite

Force recompute even if the *orid* has already been processed (`0` = no, `1` = yes).

Log Parameters

Log parameters define the location and content of the log files.

logprivate

Log database connection information (passwords) (`0` = no, `1` = yes). Extreme care should be used when using this option because database passwords are written to the log files when *logprivate*=`1`.

logargs

Write key arguments to the log: *start-time*, *end-time*, *account*, script bin directory, par file directory, and *out_origin_table* name (`0` = no, `1` = yes).

logquery

Write SQL commands to the log file prior to execution (`0` = no, `1` = yes).

logentry

Log information regarding major routines such as *EvLoc* and *maxsurf* (0 = no, 1 = yes).

logsubs

Log major script milestones (0 = no, 1 = yes).

logmaxsurf

Write all *maxsurf* generated messages to the log file (0 = no, 1 = yes).

logEvLoc

Write all *EvLoc* generated messages to the log file (0 = no, 1 = yes).

Example Par File

An example par file follows:

```
#       %G%     %W%
# PAR file for MsInterval and MsOrid
#
# Database information
account=LATEDB
out_arrival_table=OUT_ARRIVAL
out_assoc_table=OUT_ASSOC
out_netmag_table=OUT_NETMAG
out_origerr_table=OUT_ORIGERR
out_origin_table=OUT_ORIGIN
out_parrival_table=OUT_PARRIVAL
out_stamag_table=OUT_STAMAG
out_amplitude_table=OUT_AMPLITUDE
#
# Valid Phase/Magnitude types
magtype='ms_ave'
phaselist='LR'
amptype='ANL/2','LR'
#
# Stations excluded from Processing
excludelist='ASAR','TXAR'
```

```
#
# Stations for which bz channel is used
bzonly_list='HIA','BJT'
#
#
# Do not force recompute if orid already processed
overwrite=0
#
# Logging verbosity
logprivate=0  #logprivate=1 logs DB connect info.
logargs=1     #Log key command arguments
logquery=0    #Log queries
logentry=0    #Log entry and exit to Ms scripts
logsubs=0     #Log major routines within Ms scripts
logmaxsurf=1  #Log output of maxsurf
logEvLoc=1    #Log output of EvLoc
#
```

maxsurf

The *maxsurf* par file contains over 100 parameters. Definitions for each of the parameters are provided in the *maxsurf* man pages. These definitions are grouped by function in the sections that follow.

I/O Parameters

I/O parameters define the input and output data files, database tables, and attribute values.

in_db

Input database *name/password@instance* that contains the input **wfdisc** table. Either *in_db* or *in_file* is required but not both. (*libwfm*)

in_file

Directory/filename that contains the input **wfdisc** records. Either *in_file* or *in_db* is required but not both. (*libwfm*) *in_table*

Database table name where input **wfdisc** tuples may be found. Default: `wfdisc`. Optional. (*libwfm*)

origin_db

Database containing origin and station information. Use this parameter only if the data are specified through a wfdisc file instead of a table so that *in_db* is not specified. No default.

origin_table

Database table containing origin information. Default: `origin`. See *orid*.

in_affiliation

Table name for obtaining stations affiliated with a given network. Default: `affiliation`. Optional. See *net*. (*libwfm*)

instrument_table

Database table containing instrument information. Default: `instrument`.

sensor_table

Database **sensor** table containing instrument information. Default: `sensor`.

site_table

Database **site** table containing station information. Default: `site`.

sitechan_table

Database **sitechan** table containing instrument orientation information. Default: `sitechan`.

newinstdir

Directory of an instrument file to replace the recording instrument for magnitude measurement. Optional. See *newinstfile*.

newinstfile

Instrument filename to replace the recording instrument for amplitude calculations. This filename must be used for broadband data for the amplitude to be meaningful. Optional. See *newinstdir*.

out_amplitude_table

Database **amplitude** table to which results are written. If the table is not specified, amplitudes are not written to the database. If *out_amplitude_table* is specified, **parrival** and **arrival** tables must also be specified to avoid missing database links. Amplitudes are written for both noise and signal measurements. No default. Optional.

out_arrival_table

Database **arrival** table to which results are written. If the table is not specified, arrivals are not written to the database. Arrivals are written only for signal measurements, not noise measurements. No default. Optional.

out_assoc_table

Database **assoc** table to which results are written. If the table is not specified, assoc rows are not written to the database. Assoc rows are written only for signal measurements, not noise measurements. No default. Optional.

out_parrival_table

Database **parrival** table to which results are written. If the table is not specified, parrival rows are not written to the database. Parrival rows are written for both noise and signal measurements. No default. Optional.

results_dir

Directory to which a machine-readable summary of results is written (if specified). Optional. No default.

keep_old

When positive, old database entries are not deleted prior to adding new ones. Default: `0`.

noise_amptype

Amplitude type used as **amplitude**.*amptype* for a noise arrival. Default: `ANL/2`. See *noise_phase* and *write_noise*.

noise_phase

Noise phase name used in **parrival**.*phase* entries for a noise arrival. Default: `LR`. See *noise_amptype* and *write_noise*.

write_noise

When positive, a noise arrival is written to the database if a surface wave is not identified but the measured amplitude is greater than zero. Default: `0`. See *noise_phase* and *noise_amptype*.

amptype

Amplitude type that is placed in **amplitude.***amptype*. Default: `ALR/2`.

arrival_auth

Author entry that is placed in **arrival**.*auth*. Default: program name and version number.

arrival_name

Phase name that is placed in **arrival**.*iphase* and **assoc**.*phase*. Default: `LR`.

out_station_name

Alternate station name to be used for output results. This option can be used to replace a station name such as ARE0 with an array name such as ARCES, or to replace an array reference station name such as CM31 with an array name such as CMAR. The recommended procedure is to specify net or *net_list* instead of *out_station_name*. No default. Optional.

vmodel

Velocity model name to place in **assoc.***vmodel* and **parrival.***vmodel*. Default: `“-"`.

Data Selection Parameters

Data selection parameters are used to define the data to process.

orid

Origin ID of the event to be analyzed. No default. Required.

net

Name of the network that is used to obtain the stations of interest from the **affiliation** table. If only a single station in **affiliation** is linked to *net*, then *net* becomes the output station name. If multiple stations are associated with the *net*, then they are treated as an array. For long-period arrays, *net* or *sta_list* must define the elements of the array to be processed. Either *net* or *sta_list* is required but not both. See *net_list*. (*libwfm*)

net_list

List of station or array names that are treated as described in *net*. *net_list* may contain a mixture of 3-C stations and arrays. Because of the *libwfm* requirement that either *sta_list* or *net* be set, if using *net_list*, also set *net* = `DUMMY`, and do not use *sta_list*. *net* is not used if *net_list* is given.

sta_list

List of stations of interest to be used in the **wfdisc** query. Either *sta_list* or *net* is required but not both. For long-period arrays, *sta_list* must correspond to the elements of the array to be processed. (*libwfm*)

chan_list

List of channels of interest to be used in the **wfdisc** query. Required. Must be a 3-C list to use polarization filtering, for example: `'lz','ln','le'`. (*libwfm*)

deltamin

Minimum distance (degrees) for which surface waves are processed. Default: `0.0`.

deltamax

Maximum distance (degrees) for which surface waves are processed. Default: `180.0`.

gvmax

Maximum group velocity (km/s). Defines the start of the time window to be processed. Default: `5.0`.

gvmin

Minimum group velocity (km/s). Defines the end of the time window to be processed. Default: `2.0`.

start_time

Beginning epoch time of the waveform window. As with duration, a value is required, but not used. (*libwfm*)

duration

Desired time length (s) from the *start_time*. A value is required because of data checks by the *libwfm* waveform routines; however, this parameter is superseded by the group velocity window parameters and is not used. (*libwfm*)

wfd_to_wfm

Boolean flag that if positive indicates that the waveforms are to be stored in memory. This parameter must be true and cannot be omitted. (*libwfm*)

read_by_station

If a long station list or network is given as input, *libwfm* reads all of the data into memory. Particularly with broadband data, this may cause the program to grow to a very large size. Setting this flag to `1` forces the program to read and process one station at a time. With this option a new query must be performed for each station. Optional. Default: `1`.

Data Preparation Parameters

Data preparation parameters define how to prepare the data for processing.

join

Set to `1` to connect adjacent waveform segments. Time gaps are filled with values defined by *fill_value*. (*libwfm*)

fill_value

Value put in the new waveform everywhere a gap occurs. Must be set to `0.0` to fill any small gaps in the data. (*libwfm*)

fractaper

Fraction of time series to taper prior to processing. Tapering is applied to both ends of the time series. Default: `0.025`.

demean

When positive, the data are demeaned prior to processing. Default: `1`.

detrend

When positive, the data are detrended prior to processing. Default: `1`.

azshift

Expected azimuth shift (degrees) for a given path. That is, if an arrival consistently comes in 10 degrees off for a given path, *azshift* should be set to `10`. Default: `0.0`.

Test Parameters

Test parameters define how the data are tested.

NumFailAllowed

Set to non-zero to allow for a fixed number of test failures. Default: `0`.

fdisp

Array of frequencies (Hz) corresponding to the group velocity dispersion in *vdisp*. Limited to 100 values. Default not used. All values must be greater than zero. Not needed if *PCORregdir* is given. See *vdisplimit*, *pdisplimit*, and *tdisplimit*.

vdisp

Array of group velocities (km/s) that specify the predicted dispersion. Must be used together with *fdisp*. Limited to 100 values. Default not used. All values must be greater than zero. Not needed if *PCORregdir* is given. See *vdisplimit*, *pdisplimit*, and *tdisplimit*.

pdisplimit

Acceptable range for the number of periods that arrival times may be outside of the predicted dispersion curve. Arrivals are limited to the time window specified in *NBFtestDispersion*. Default: `1.0`. See *vdisplimit*, *tdisplimit*, and *vdisp*.

tdisplimit

Acceptable range of arrival times (s) outside of the predicted dispersion curve. Arrivals are limited to the time window specified in *NBFtestDispersion*. Default: `0.0`. See *vdisplimit* and *pdisplimit*.

vdisplimit

Acceptable group velocity (km/s) range for which arrival times may be outside of the predicted dispersion curve. Arrivals are limited to the time window specified in *NBFtestDispersion*. Default: `0.2`. See *pdisplimit* and *tdisplimit*.

fstatmin

Minimum allowed F-statistic for polarization filtering and long-period array processing. Default: `1.0`. See *POLtestFstat* and *LPAtestFstat*.

azdiffmax

Maximum allowed difference (degrees) between the apparent azimuth and the actual back azimuth in the polarization and long-period array azimuth tests. Default: `30.0`.

elmin

Minimum allowed value for the ellipticity test. Default: `0.4`. See *elmax* and *POLtestEllip*.

elmax

Maximum allowed value for the ellipticity test. Default: `2.0`. See *elmin* and *POLtestEllip*.

override_accept

When positive, the surface wave arrival is accepted and written to the database even if it fails the surface wave acceptance tests. This option is provided for those cases where an analyst can identify a surface wave that for one reason or another fails the tests applied by *maxsurf*. Default: `0`.

override_reject

When positive, the surface wave arrival is rejected and not written to the database (while removing it if it has been added previously) even if it passes the surface wave acceptance tests. This option is provided for those cases where an analyst believes that the surface wave tests were passed fortuitously and a surface wave from the event being analyzed does not actually exist in the time window. Default: `0`.

Instrument Response Parameters

Instrument response parameters define how to correct the data for instrument response prior to processing.

instfmax

Upper frequency (Hz) limit for instrument response corrections. Data are set to zero at higher frequencies. Instrument response corrections are performed for array analysis, narrow-band filtering, polarization filtering, and magnitude measurement. This option can save considerable CPU time with broadband data that have not been decimated. Default: `0.5`.

newinstwaterlevel

Lowest meaningful amplitude level (digital counts/nanometer) of old instrument added to old response before dividing. Default: `1.e-4`.

newinstcalib

Amplitude (nanometers/digital count) of the new instrument at the calibration period. See *newinstcalper*.

newinstcalper

Calibration period (s) of the new instrument. If not given, the response is not scaled. See *newinstcalib*.

Decimation Parameters

Decimation parameters define how the input data are decimated prior to processing.

DECfilters

List of decimation filters to be applied to the data. Decimation filters are chosen from `SAC2`, `SAC3`, `SAC4`, `SAC5`, `SAC6`, and `SCAT4`, where `SAC[2–6]` are decimation filters derived from the Seismic Analysis Code (SAC) [Gol96] and represent decimation by factors of 2–6, respectively. `SCAT4` is a shorter, faster filter for decimation by 4. Default: (see *DECsamprate*).

DECsamprate

Desired sampling rate (samples/s). *maxsurf* uses the decimation filters (see *DECfilters*) to come close to this sampling rate. Default: `1`.

Long-period Array Parameters

Long-period array (LPA) parameters define the processing for long-period array data.

LPArray

Positive values indicate that the input data are to be treated as an array. All stations used for array processing must have the same reference station listed in the **site** table (defined by parameter *site_table*). By default, the station name used for array results is the reference station. The station name may be changed by specifying *out_station_name*. Default: `0`.

LPAazmax

Maximum azimuth value (degrees) used in the LP array beam search. Default: `360.0`.

LPAazmin

Minimum azimuth value (degrees) used in the LP array beam search. Default: `0.0`.

LPAaznum

Number of azimuth values used in the LP array beam search. Default: `73` (every 5 degrees).

LPAbeam3c

When positive, 3-C beaming is used for further processing. When `0`, the vertical component only is beamed. Default: `0`.

LPAbeamEllip

Ellipticity used to form the long-period beam for further processing. This should be the ellipticity corresponding to the earth structure near the array, not the path. A constant (frequency-independent) value should be adequate. Used only for 3-C beamforming. Default: `0.8`. Beam ellipticity and phase velocity can also be specified as frequency-dependent arrays of type float using the input parameters *LPAbeamFreqArr*, *LPAbeamEllipArr*, and *LPAbeamVelArr*.

LPAbeamEllipArr

Frequency-dependent ellipticity used to form the long-period beam for further processing. Each member of the comma separated list corresponds to the frequency set by *LPAbeamFreqArr*. The ellipticity corresponds to the earth structure near the array, not the path. See *LPAbeamVelArr*.

LPAbeamFreqArr

Frequencies (Hz) used in frequency-dependent ellipticity and phase velocity specifications (comma separated) . See *LPAbeamEllipArr* and *LPAbeamVelArr*.

LPAbeamRegional

When positive, a regionalized phase velocity model is used for beamforming instead of a fixed phase velocity. See *LPAbeamVel* and *LPAbeamVelArr*. The phase velocity is specified in the same format as the regionalized dispersion curve in directory *PCORregdir*, with name `LP_pvel.LR` specified by the grid file `LP_grid.LR`. The two files may be combined in binary form as `LP_pvel.LR.o`. Default: `0`.

LPAbeamVel

Phase velocity (km/s) used to form the long-period beam used for further processing. This should be the phase velocity across the array, not along the path. A constant (frequency-independent) value should be adequate. Default: `3.5`. Beam phase velocity and ellipticity can also be specified as frequency-dependent arrays of type float using the input parameters *LPAbeamFreqArr*, *LPAbeamEllipArr*, and *LPAbeamVelArr*.

LPAbeamVelArr

Frequency-dependent phase velocity (km/s) used to form the long-period beam for further processing. Each member of the comma separated list corresponds to the frequency set by *LPAbeamFreqArr*. The phase velocity is across the array, not along the path. See *LPAbeamEllipArr*.

LPAdelStations

When positive, any arrivals that exist in the database for individual stations in the array as well as arrivals for the reference stations are deleted. Default: `1`.

LPAelmax

Maximum ellipticity value used in the LP array beam search. Default: `0.8`.

LPAelmin

Minimum ellipticity value used in the LP array beam search. Default: `0.8`.

LPAelnum

Number of ellipticity values used in the LP array beam search. Default: `1`.

LPAfreq

Set of comma separated frequencies (Hz) used to form the beam for the long-period array azimuth search. Default: `0.02`, `0.025`, `0.03`, `0.035`, `0.04`, `0.05`, `0.06`. All values must be greater than zero.

LPAhorWeight

Sets the weighting of all horizontal component data used in array processing. If set to `0`, only vertical component data are used. If *LPAbeam3c* is set to `0` while *LPAhorWeight* is positive, then 3-C data are used for the azimuth search, but only vertical component data are used for forming the beam for additional processing. Default: `1.0`.

LPAverWeight

Sets the weighting of all vertical component data used in array processing. *LPAverWeight* and *LPAhorWeight* are scale independent. That is, for example, weights of `10.0` and `1.0`, respectively, give the same results as `1.0` and `0.1`. Default: `1.0`.

LPAtestAz

When positive, the long-period array test for measured azimuth close to the actual back azimuth is applied. Default: `0`. See *azdiffmax* and *azshift*.

LPAtestFstat

When positive, the long-period array test for minimum F-statistic is applied. Default: `0`. See *fstatmin*.

LPAvpmax

Maximum phase velocity (km/s) used in the LP array beam search. Default: `4.5`.

LPAvpmin

Minimum phase velocity (km/s) used in the LP array beam search. Default: `2.5`.

LPAvpnum

Number of phase velocity values used in the LP array beam search. Default: `21`. The LP array beam data are searched on even intervals of slowness between 1/*LPAvpmax* and 1/*LPAvpmin*.

Narrow-band Filter Parameters

Narrow-band filter (NBF) parameters define filter processing.

NBFdo

When positive, narrow-band filters are applied to measure group velocity dispersion. Default: `1`.

NBFfilterQ

Filter Q used for narrow-band filter analysis. Default: `15.0`.

NBFfreq

Set of frequencies (Hz) used for narrow-band filtering. Default: `0.02`, `0.025`, `0.03`, `0.035`, `0.04`, `0.045`, `0.05`, `0.06`. The default frequencies should always be used for *NBFtestNorth*. All values must be greater than zero.

NBFnoinst

When positive, the instrument group delay from narrow-band filtering is not corrected. This is a time-consuming operation and a small correction if the instrument phase is slowly varying. Default: `0`.

NBFresultsdir

Directory to write predicted and observed group velocity dispersion curves for each event/station pair. No default. If not given, dispersion curves are written to standard output.

NBFtestDispersion

When positive, the group velocity arrival times are tested for consistency with predicted group velocities. The group velocity arrival times are determined by narrow-band filtering. The predicted arrival time may either be specified as input frequencies and velocities *fdisp* and *vdisp* or derived from a regionalized group velocity model (see *PCORregdir*). The expected arrival time interval is specified by defining a time window around the predicted group velocity time. This may be specified as a group velocity range, time range, or period-dependent range using the parameters *vdisplimit*, *tdisplimit*, and *pdisplimit*, respectively. The measured group arrivals are required to be in the time interval:

$$\mathrm{tmin} = \frac{\text{distance}}{(vdisp + vdisplimit)} - (pdisplimit \cdot \text{period}) - tdisplimit$$

and

$$\mathrm{tmax} = \frac{\text{distance}}{(vdisp - vdisplimit)} + (pdisplimit \cdot \text{period}) + tdisplimit$$

The default values of these parameters are *vdisplimit* = `0.2`, *pdisplimit* = `1.0`, and *tdisplimit* = `0.0`. For example, if the expected group velocity of a 20 s arrival is 3.0 km/s and the distance from event to station is 6,000 km, the default time window is from 1,855 to 2,163 s after origin time. A surface wave passes this test if the fraction of points determined at different frequencies satisfying this condition exceeds *NBFtestDispFrac*. See *NBFtestDispFrac*, *fdisp*, *vdisp*, *vdisplimit*, *pdisplimit*, and *tdisplimit*. Default: `0`.

NBFtestDispFrac

Fraction of narrow-band filter group velocity estimates that must lie within the predicted arrival time window to pass the dispersion test. See *NBFtestDispersion*, *fdisp*, *vdisp*, and *vdisplimit*. Default: `0.7`.

NBFtestNorth

When positive, the North and Woodgold [Nor94] test for surface wave dispersion is applied. This test looks for decreasing group velocity with increasing frequency and allows for one exception at the lowest or highest frequency and one exception in between. Default: `0`.

Polarization Parameters

Polarization parameters define if and how to process the data using polarization filtering.

POLdo

When positive, polarization filtering to estimate station to event azimuth is applied. Default: `1`.

POLfmax

Ending frequency (Hz) for polarization filtering. Default: `0.05`.

POLfmin

Starting frequency (Hz) for polarization filtering. Default: `0.02`.

POLnoinst

When non-zero, the instrument response correction is not applied prior to polarization filtering. This is primarily an optimization feature, because instrument correction can be quite time consuming in some cases. Default: `0`.

POLtestAz

When positive, the polarization test for measured azimuth close to actual back azimuth is applied. Default: `0`. See *azdiffmax*.

POLtestFstat

When positive, the test for F-statistic of the polarization test is applied. Default: `0`. See *fstatmin*.

POLtestEllip

When positive, the test for reasonable ellipticity in the polarization test is applied. Default: `0`. See *elmin* and *elmax*.

POLtestAll

When positive, *POLtestAz*, *POLtestFstat*, and *POLtestEllip* are combined into a single test.

Path-correction Parameters

Path-correction (PCOR) parameters define the location of the dispersion files.

PCORcompiled

When positive, regionalized dispersion curves are read in binary rather than ASCII format. The file `LP_vel.LR.o` must exist in the directory *PCORregdir*. This can lead to a substantial savings in time for detailed dispersion models. The files can be compiled to binary form with the program *LPcompile*. Default: `0`.

PCORregdir

Directory containing regionalized group velocity curves. The directory must contain two files named `LP_grid.LR` and `LP_vel.LR`, which specify grid points and group velocities for Rayleigh waves. The two files may be combined in binary form as `LP_vel.LR.o`. Optional. No default.

Magnitude Parameters

Magnitude parameters define if and how surface wave magnitudes are calculated.

MAGdo

When positive, *maxsurf* searches for a surface wave arrival and calculates a magnitude. If *MAGdo* is not positive, no arrival is found and surface wave identification and measurement is not performed. Default: `1`.

MAGBtest

Prunes out unreasonable surface wave amplitudes based on two tests against the m_b for the event. This can only be applied if m_b is defined in the **origin** table. The first test is a simple threshold test based on magnitude and distance, rejecting an arrival if

$m_b < MAGBmin + 1.66 \log(\Delta/MAGBdelta)$

where the default values of *MAGBmin* and *MAGBdelta* are `2.5` and `20.0`, respectively. The second test removes arrivals where M_s is larger than all historical values of M_s for events of the same m_b. Arrivals are removed if M_s exceeds $m_b + 2$ for $m_b < 5$, $m_b + 1.5$ for $m_b < 4.5$, and $m_b + 1$ for $m_b < 4$.

MAGBdelta

Distance (degrees) corresponding to *MAGBmin* if *MAGBtest* is applied. Default: `20.0`.

MAGBmin

Minimum magnitude (m_b) at *MAGBdelta* for which a surface wave arrival is allowed. Default: `2.5`. See *MAGBtest*.

MAGdfac

"D" in the equation for M_s, as defined by Rezapour and Pearce [Rez98]:

$$M_s = \log\left(\frac{A}{T}\right) + k \cdot \log(\Delta) + 0.5 \cdot \log(\sin(\Delta)) + \gamma \cdot \Delta \cdot \log(e) + D$$

where k is 1/3 or 1/2 depending on whether the arrival is an Airy phase or normally dispersed. The Rezapour and Pearce values are $k = 1/3$, $\gamma = 0.0105$, $D = 2.370$. These values are set with the parameters *MAGkfac*, *MAGgamma*, and *MAGdfac*, respectively, which default to the Rezapour and Pearce values. Default: `2.370`.

MAGgamma

"γ" in the equation for M_s, as defined by Rezapour and Pearce [Rez98].

$$M_s = \log\left(\frac{A}{T}\right) + k \cdot \log(\Delta) + 0.5 \cdot \log(\sin(\Delta)) + \gamma \cdot \Delta \cdot \log(e) + D$$

where *k* is 1/3 or 1/2 depending on whether the arrival is an Airy phase or normally dispersed. The Rezapour and Pearce values are *k* = 1/3, γ = 0.0105, D = 2.370. These values are set with the parameters *MAGkfac*, *MAGgamma*, and *MAGdfac*, respectively, which default to the Rezapour and Pearce values. Default: `0.0105`.

MAGkfac

"*k*" in the equation for M_S, as defined by Rezapour and Pearce [Rez98]:

$$M_s = \log\left(\frac{A}{T}\right) + k \cdot \log(\Delta) + 0.5 \cdot \log(\sin(\Delta)) + \gamma \cdot \Delta \cdot \log(e) + D$$

where *k* is 1/3 or 1/2 depending on whether the arrival is an Airy phase or normally dispersed. The Rezapour and Pearce values are *k* = 1/3, γ = 0.0105, D = 2.370. These values are set with the parameters *MAGkfac*, *MAGgamma*, and *MAGkfac*, respectively, which default to the Rezapour and Pearce values. Default: `0.33333`.

MAGtest

When positive, a test for reasonableness of magnitude is applied. Default: `0`.

MAGmax

Maximum allowed value (M_S) for magnitude test. Default: `10.0`. See *MAGmin* and *MAGtest*.

MAGmin

Minimum allowed value (M_S) for magnitude test. Default: `2.0`. See *MAGmax* and *MAGtest*.

MAGperMin

Minimum allowable period (s) for a magnitude measurement. Default: `10.0`.

MAGperMax

Maximum allowable period (s) for a magnitude measurement.
Default: `30.0`.

MAGperPref

Preferred period (s) for a magnitude measurement.
Default: `20.0`.

MAGperSlope

Period weighting function to be applied to determine the peak that corresponds to the desired arrival. The weighting function has the form:

$$W = MAGperSlope \cdot \mathrm{ABS}(\mathrm{period} - MagperPref)$$

The arrival is then found as follows: A magnitude is calculated for each peak in the waveform satisfying the arrival time test (see *vdisp*) and having a period between *MAGperMin* and *MAGperMax*. The weighting function *W* is then subtracted from each magnitude, and the largest weighted value is assumed to correspond to the surface wave arrival. Default: `0.2`. The default value is weighted rather strongly toward 20 s to identify the best 20 s arrival. If *MAGperSlope* is set to zero, then there is no weighting, and the maximum value between *MAGperMin* and *MAGperMax* is set as the surface wave arrival. See *vdisplimit*.

MAGtype

Surface wave magnitude type. *MAGtype* must be one of `"MsGutenberg"`, `"MsPrague"`, `"MsVonSeggern"`, `"MsMarshall-Basham"`, `"MsTheoretical"`, or `"MsRP"`, which correspond to the original Gutenberg magnitude [Aki80], the Prague (IASPEI) variant of Gutenberg's magnitude [Aki80], Von Seggern's M_S [von77], which has an attenuation correction of 1.0 instead of 1.66, the Marshall and Basham's M_S [Mar72] based on theoretical Rayleigh wave attenuation, and M_S as defined by Rezapour and Pearce [Rez98], which uses the theoretical formula with empirical values for the constants. `MsTheoretical` has the form:

$$M_s = \log\left(\frac{A}{T}\right) + k \cdot \log(\Delta) + 0.5 \cdot \log(\sin(\Delta)) + \gamma \cdot \Delta \cdot \log(e) + D$$

where k is 1/3 or 1/2 depending on whether the arrival is an Airy phase or normally dispersed. The Rezapour and Pearce values are $k = 1/3$, $\gamma = 0.0105$, D = 2.370. For `MsTheoretical`, these values are set with the parameters *MAGkfac*, *MAGgamma*, and *MAGdfac*, respectively, which default to the Rezapour and Pearce values. The Marshall Basham magnitude corrects for path type and depth as well as distance. Default: `"MsPrague"`. See *MAGpathtype*.

MAGpathtype

Path type used for the Marshall Basham magnitude. It must be one of `"Continental"`, `"Oceanic"`, `"Mixed"`, or `"Eurasia"`. This parameter applies a path-dependent period correction to the magnitude. Not used for other magnitudes. Default: `Continental`.

Diagnostic Parameters

Diagnostic parameters are used to evaluate the performance of *maxsurf*.

Timer

When positive, the time that each part of the analysis takes is printed to the standard output, which is usually redirected to a log file. Default: `0`.

Example Par File

An example of a *maxsurf* par file follows:

```
# @(#)maxsurf.par        1.1 04/21/98
#
par=$(IMSPAR)
par=$(AUTOMATIC)
in_db=$(LATEDB)

# print out time spent in various modules
Timer=1
```

```
LPAbeamEllip=0.8
LPAbeamVel=3.5
LPAbeam3c=0
LPAhorWeight=1.0
LPAverWeight=1.0

#MAGB parameters
MAGBtest=1
MAGBdelta=20.0
MAGBmin=2.5

#MAG parameters
MAGmax=10.0
MAGmin=2.0
MAGperSlope=0.
MAGperMin=18.
MAGperMax=22.
MAGtest=1

#NBF parameters
NBFfilterQ=15
NBFfreq=0.02,0.025,0.03,0.035,0.04,0.045,0.05,0.06
NBFtestDispersion=1

NumFailAllowed=0

# compiled dispersion curves
PCORregdir=$(STATICDIR)/LPdisp/LP
PCORcompiled=1

POLtestAll=0
azdiffmax=45.0
arrival_name=LR

deltamax=100.0
```

```
deltamin=0.0
demean=1
detrend=1
duration=100000.0

fill_value=0.0
gvmax=5.0
gvmin=2.0
in_affiliation=affiliation
in_table=wfdisc
instrument_table=instrument
join=1

# net or sta_list must be given even though not used
net=DUMMY
newinstcalib=1.0
newinstcalper=20.
newinstdir=$(STATION-DIR)/rsp
newinstfile=KS36000.lp

origin_table=origin
out_arrival_table=arrival
out_assoc_table=assoc
out_amplitude_table=amplitude
out_parrival_table=parrival
sensor_table=sensor
site_table=site
sitechan_table=sitechan
start_time=0.0

pdisplimit=1.0
tdisplimit=0.0
vdisplimit=0.2

wfd_to_wfm=1
```

```
# write noise to database
write_noise=1

# set amptypes
amptype=ALR/2
noise_amptype=ANL/2

# keep old LR's. (MsOrid deletes them, so it is safe to
# "keep" them here.
keep_old=1
```

DATABASE

This section describes database elements required for operation of this software component, including accounts, tables, and initialization of the **lastid** table.

Accounts

The SWIM software requires the LEB database account with read and write permissions. Some of the tables accessed are synonyms for tables in the IDCX and STATIC accounts.

Tables

The SWIM software uses the **affiliation**, **amplitude**, **arrival**, **assoc**, **instrument**, **netmag**, **origin**, **origerr**, **parrival**, **sensor**, **site**, **sitechan**, **stamag**, and **wfdisc** tables. The table names are defined in the parameter files and may differ from these as long as they have the same structure.

lastid Initialization

The *maxsurf* program writes to the **arrival**, **assoc**, **parrival**, and **amplitude** tables, and key ID values for these tables are obtained from the **lastid** table. The key names are *ampid*, *arid*, and *parid*. Key ID values are requested with the *gdi_get_counter* function.

TUXEDO FILES

The *MsInterval* and *MsOrid* scripts are started by *tuxshell*, which is a component of the DACS. Parameter files, configuration entries in the `ubb_process.tmpl` file, and an established queue space are required for proper operation of the SWIM software.

Par Files

The SWIM software script *MsInterval* is started by the DACS as part of recall processing. A *tuxshell* par file must be included in the recall processing Tuxedo configuration. At the PIDC, this file (`tuxshell-MsOrid.par`) is located in `/cmss/config/app_config/distributed/tuxshell/recall/`. The following par file is used at the PIDC:

```
#
#       batch MsOrid computations
#

role=MsOrid
prefix=Ms

par=$(IMSPAR)
par=$(DISTRIBUTED)

sendkeys=0

Ms-exec=$(SCRIPTSBIN)/MsInterval

Ms-key[0]=time
Ms-key[1]=endtime

#  Put these values on the command line.
Ms[1]="start-time=$(time)"
Ms[2]="end-time=$(endtime)"
Ms[3]="par=$(AUTOMATIC-DIR)/MsInterval/MsInterval.par"

Ms-timeout=7200
Ms-true-exit=0

# Queuing Flow
#
interval-source=queue
```

```
destqueue=done

interval-success-state=done

log=$(LOGDIR)/%jdate/tuxshell/$(role)-%host-%pid
outfile=$(LOGDIR)/%jdate/Recall/$(role)-%host-%pid
errfile=$(outfile)
```

The SWIM software script *MsOrid* may be started during interactive analysis for specific origins through the DACS. A *tuxshell* par file must be included in the interactive processing Tuxedo configuration. At the PIDC, this file (`tuxshell-MsOrid.par`) is located in `/cmss/config/app_config/distributed/tuxshell/interactive/`. The following example is the par file used at the PIDC:

```
# @(#)tuxshell-MsOrid.par       1.1 04/21/98
#

role=MsOrid
prefix=$(role)

par=$(IMSPAR)
par=$(DISTRIBUTED)

agent=INTERACTIVE
qspace=INTERACTIVE
mode=interactive
send-reply-message=0

# Override of value set in dacs.par
field=

return-msg-id="EndMsOrid"
return-msg-ok="action=Save"
return-msg-err="action=Cancel"

MsOrid-exec="$(RELBIN)/MsOrid"

sendkeys

MsOrid-key[0]=orid
MsOrid[1]="par=$(AUTOMATIC-DIR)/MsOrid/MsOrid.par

MsOrid-true-exit=0
```

```
interval-source=queue

log=$(LOGDIR)/%jdate/tuxshell/$(role)-%host-%pid
outfile=$(LOGDIR)/%jdate/interactive/$(role)-%host-%pid
errfile=$(outfile)
```

ubbconfig Template

To ensure that Tuxedo is configured properly for SWIM software operation, verify that the necessary entries listed below are present in the file `ubb_process.tmpl`. At the PIDC this file is in the directory `/cmss/config/system_specs/`.

```
tuxshell        SRVGRP=RCL_PRI          SRVID=950
   CLOPT="-s MsOrid:tuxshell -o /dev/null -e /dev/null --
par=/cmss/config/app_config/distributed/tuxshell/recall/tuxshell-
MsOrid.par"
   REPLYQ=Y        RQADDR="MsOrid"         MIN=3   MAX=6

tuxshell        SRVGRP=RCL_BAK          SRVID=10950
   CLOPT="-s MsOrid:tuxshell -o /dev/null -e /dev/null --
par=/cmss/config/app_config/distributed/tuxshell/recall/tuxshell-
MsOrid.par"
   REPLYQ=Y        RQADDR="MsOrid-bak"      MIN=3   MAX=6

TMQFORWARD      SRVGRP=QM_PRI   SRVID=5950
   CLOPT="-- -i 10 -q MsOrid   -t 7300"     MIN=3   MAX=6

TMQFORWARD      SRVGRP=QM_BAK   SRVID=15950
   CLOPT="-- -i 10 -q MsOrid   -t 7300"     MIN=3   MAX=6

MsOrid          LOAD=10         SRVGRP=RCL_PRI
MsOrid          LOAD=20000      SRVGRP=RCL_BAK
```

Queue Space

During the DACS configuration, the script *crDacsQueues* is used to create queue space for all software subsystems under Tuxedo control. To ensure that Tuxedo is configured properly for SWIM software operation, verify that the necessary entry listed below is present in the script *crDacsQueues*. At the PIDC this script is located in directory `/cmss/scripts/bin/`.

```
qcreate MsOrid       priority,time top,msgid 2 30 80% 0%
"$CMS_SCRIPTS/bin/mailFullQ MsOrid"
```

If the entry needed to create the *MsOrid* queue space is not present in the *crDacsQueues* script, you must add it. However, this should not be necessary because the *crDacsQueues* script is released to the IDC with all necessary entries.

Queue spaces can only be created during configuration of the Tuxedo system. To create a new *MsOrid* queue space (or queue space for any software), halt IDC system processing, erase existing queue spaces, create new queue spaces with the amended *crDacsQueues* script, and restart IDC processing. For more information on configuring Tuxedo, refer to [IDC6.5.2Rev0.1].

ARS SCHEME

Analysts may calculate surface wave magnitudes using *ARS*, which must be configured to send a message to start *MsOrid* for a specific origin using the DACS. *ARS* is configured using Scheme. At the PIDC the Scheme configuration file (`IDC.scm`) is in `/cmss/config/app_config/interactive/ARS/`. The following lines of this Scheme file define the *MsOrid* message:

```
;-->   send-MS-orid-message
;;;
;;;    used by
;;;      (add-button "Toolbar" "MS" "(send-MS-orid-message)")
;;;      ARS*sendMenu*send-item6.eval: (send-MS-orid-message)
;;;
;;;     no default

(define (send-MS-orid-message)
  (let ((origin (car (say-selected-origins))))
    (if (origin? origin)
      (let ((message (quote-string (string-append " orid="
        (get-string (extract-object-attribute origin "orid")) " "))))
        (send-ipc-message "MsOrid" "RunMsOrid" message) message
      ) (say-info
          (list "send-MS-orid-message called, but no origin selected."))
    )
  )
)
```

INITIATING OPERATIONS

After the SWIM software and files are installed, use *LPcompile* to generate the binary file of regional dispersion models. Operations are initiated automatically through the DACS; no intervention is required.

VALIDATING INSTALLATION

To validate installation, run the SWIM software on a full day of events, and then check the log files for error messages and the database to ensure that all surface waves have been recorded. In addition, check the amplitudes at each station for consistency with other stations. If consistently large residuals exist, then an incorrect instrument response file is indicated. M_s residuals due to real station differences rarely exceed 0.2 magnitude units. Check M_s residuals monthly to identify errors due to instrument response changes.

References

The following sources supplement or are referenced in the document:

[Aki80]	Aki, K., and Richards, P., *Quantitative Seismology, Theory and Methods*, Volume 1, Appendix 2, pp. 533–534, W. H. Freeman and Company, New York, 1980.
[Gan79]	Gane, C., and Sarson, T., *Structured Systems Analysis: Tools and Techniques*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1979.
[Gol96]	Goldstein, P., and Minner, L., "SAC2000: Seismic Signal Processing and Analysis Tools For the 21st Century," *Seis. Res. Lett.*, Volume 67, p. 39, 1996.
[IDC5.1.1Rev2]	Science Applications International Corporation, Veridian Pacific-Sierra Research, *Database Schema, Revision 2*, SAIC-00/3057, PSR-00/TN2830, 2000.
[IDC5.2.1]	Science Applications International Corporation, *IDC Processing of Seismic, Hydroacoustic, and Infrasonic Data*, SAIC-99/3023, 1999.
[IDC6.5.2Rev0.1]	Science Applications International Corporation, *Distributed Application Control System (DACS) Software User Manual, Revision 0.1*, SAIC-00/3038, 2000.
[IDC7.1.3]	Science Applications International Corporation, *Surface Wave Identification and Measurement*, SAIC-01/3008, 2001.
[IDC7.1.5]	Science Applications International Corporation, *Event Location Software,* SAIC-01/3010, 2001.

[Mar72] Marshall, P. D., and Basham, P. W., "Discriminating Between Earthquakes and Underground Explosions Employing an Improved Ms Scale," *Geophysical Journal of the Royal Astronomical Society*, Volume 28, pp. 431–458, 1972.

[Nor94] North, R. G., and Woodgold, C. R. D., "Automated Detection and Association of Surface Waves," *Annali de Geofisica*, Volume 37, pp. 301–308, 1994.

[Rez98] Rezapour, M., and Pearce, R. G., "Bias in Surface-wave Magnitude M_S Due to Inadequate Distance Correction," *Bulletin of the Seismological Society of America*, Volume 88, pp. 43–61, 1998.

[Sma71] Smart, E., and Flinn, E. A., "Frequency-Wave Number Analysis and Fisher Signal Detection in Real-Time Infrasonic Array Data Processing," *Geophysical Journal of the Royal Astronomical Society,* Volume 26, pp. 279–284, 1971.

[Ste86] Stevens, J. L., "Estimation of Scalar Moments from Explosion-Generated Surface Waves," *Bulletin of the Seismological Society of America*, Volume 76, pp. 123-151, 1986.

[Ste96] Stevens, J. L., *Regionalized Maximum Likelihood Surface Wave Analysis*, Maxwell Technologies Technical Report, PL-TR-96-2273, SSS-DTR-96-15562, ADA 321813, 1996.

[Ste97] Stevens, J. L., and McLaughlin, K. L., *Improved Methods for Regionalized Surface Wave Analysis*, Phillips Laboratory, Directorate of Geophysics, AFMC, Hanscom AFB, MA, PL-TR-97-2135, 1997.

[Ste01] Stevens, J. L., and McLaughlin, K. L., "Optimization of Surface Wave Identification and Measurement," *Pure and Applied Geophysics*, Volume 158, 2001.

[von77] von Seggern, D., "Amplitude-distance Relation for 20-second Rayleigh Waves," *Bulletin of the Seismological Society of America,* Volume 67, pp. 405–411, 1977.

Glossary

Symbols

3-C

Three-component.

A

amplitude

Zero-to-peak height of a waveform in nanometers.

Analyst Review Station

This application provides tools for a human analyst to refine and improve the event bulletin by interactive analysis.

arid

Arrival identifier.

array

Collection of sensors distributed over a finite area (usually in a cross, triangle, or concentric pattern) and referred to as a single station.

arrival

Detected signal that has been associated to an event. First, the Global Association (GA) software associates the detection to an event. Later, during interactive processing, many arrivals are confirmed, improved, or added by visual inspection.

ARS

See Analyst Review Station.

ASCII

American Standard Code for Information Interchange. Standard, unformatted 256-character set of letters and numbers.

azimuth

Direction, in degrees clockwise with respect to North, from a station to an event.

B

back azimuth

Direction, in degrees, from an event to a station.

beam

(1) Waveform created from array station elements that are sequentially summed after being steered to the direction of a

specified azimuth and slowness. (2) Any derived waveform (for example, a filtered waveform).

beamform

Sum a set of waveforms from array station elements with time delays introduced to compensate for the time it takes a wave to travel across the array.

bulletin

Chronological listing of event origins spanning an interval of time. Often, the specification of each origin or event is accompanied by the event's arrivals and sometimes with the event's waveforms.

C

CD-ROM

Compact Disk–Read Only Memory.

channel

Component of motion or distinct stream of data.

computer software component

Functionally or logically distinct part of a computer software configuration item, typically an aggregate of two or more software units.

computer software configuration item

Aggregation of software that is designated for configuration management and treated as a single entity in the configuration management process.

CPU

Central Processing Unit.

crash

Sudden and complete failure of a computer system or component.

CSC

See computer software component.

CSCI

See computer software configuration item.

D

DACS

See Distributed Application Control System.

dispersion

Expansion of the length of a seismic wavetrain due to each wavelength travelling with its own velocity.

Distributed Application Control System

This software supports inter-application message passing and process management.

E

element

(1) Single station or substation of a sensor array, referred to by its element name (such as YKR8), as opposed to its array name (YKA in this example). (2) Data storage location in a data array.

epoch time

Number of seconds after January 1, 1970 00:00:00.0.

event

Unique source of seismic, hydroacoustic, or infrasonic wave energy that is limited in both time and space.

event screening

IDC process that produces the SSEB by removing events of clear natural origin from the SEB.

EvLoc

Application used to compute event location and/or magnitude.

F

F-statistic

Measure that indicates the degree of spatial coherence of a waveform across an array of sensors. This measure is approximately equal to the ratio of the spatially coherent energy to the incoherent energy scaled by the number of non-collocated sensors.

FTP

File Transfer Protocol; protocol for transferring files between computers.

G

GB

Gigabyte. A measure of computer memory or disk space that is equal to 1,024 megabytes.

group velocity

Propagation velocity of packets or groups of waves.

GSETT-3

Group of Scientific Experts Third Technical Test.

H

Hz

Hertz.

I

ID

Identification; identifier.

IDC

International Data Centre.

IMS

International Monitoring System.

IPC

Interprocess communication. The messaging system by which applications communicate with each other through *libipc* common library functions. See *tuxshell*.

K

KB

Kilobyte. 1,024 bytes.

km

Kilometer.

L

Late Event Bulletin

List of analyst reviewed S/H/I events and event parameters (origin and associated arrival information).

LEB

See Late Event Bulletin.

LP

Long period.

LR

See Rayleigh wave.

M

magnitude

Empirical measure of the size of an event (usually made on a log scale).

MB

Megabyte. 1,024 kilobytes.

m_b

Magnitude estimated from seismic body waves.

M_s

Magnitude of seismic surface waves.

N

network

Spatially distributed collection of seismic, hydroacoustic, or infrasonic stations for which the station spacing is much larger than a wavelength.

network processing

Processing that uses the results of Station Processing from a network of stations to define and locate events.

noise

Incoherent natural or artificial perturbations of the waveform trace caused by ice, animals migrations, cultural activity, equipment malfunctions or interruption of satellite communication, or ambient background movements.

O

ORACLE

Vendor of the database management system used at the PIDC and IDC.

origin

Hypothesized time and location of a seismic, hydroacoustic, or infrasonic event. An event may have many origins. Characteristics such as magnitudes and error estimates may be associated with an origin.

P

par

See parameter.

parameter

User-specified token that controls some aspect of an application (for example, database name, threshold value). Most parameters are specified using [*token* = *value*] strings, for example, `dbname=mydata/base@oracle`.

parameter (par) file

ASCII file containing values for parameters of a program. Par files are used to replace command line arguments. The files are formatted as a list of `[`*token* `=` *value*`]` strings.

period

Average duration of one cycle of a phase, in seconds per cycle.

phase

Arrival that is identified based on its path through the earth.

PIDC

Prototype International Data Centre.

polarization

Form of three-component analysis used to derive azimuth and slowness information from non-array stations.

polarization analysis

Analysis to determine the propagation vector that describes particle motion. Used at seismic 3-C stations.

post-location processing

Software that computes various magnitude estimates and selects data to be retrieved from auxiliary stations.

primary stations

Stations that make up the primary seismic network of the IMS. Primary stations send data continuously to the IDC.

Q

Q

Dimensionless quality factor inversely related to the strength of attenuation of seismic wave amplitudes.

query

Request for specific data from a database.

R

radionuclide

Pertaining to the technology for detecting radioactive debris from nuclear reactions.

Rayleigh wave

A seismic phase that travels along the surface of the earth.

REB

See Reviewed Event Bulletin.

regional

(1) (distance) Source-to-seismometer separations between a few degrees and 20 degrees. (2) (event) Recorded at distances where the first P and S waves from shallow events have traveled along paths through the uppermost mantle.

residual

Difference between the observed value for an attribute (for example, time, azimuth, slowness, or magnitude) and its corresponding theoretical value.

Reviewed Event Bulletin

Bulletin formed of all S/H/I events that have passed analyst inspection and quality assurance review.

S

s

Second(s) (time).

S/H/I

Seismic, hydroacoustic, and infrasonic.

SAIC

Science Applications International Corporation.

script

Small executable program, written with UNIX and other related commands, that does not need to be compiled.

seismic

Pertaining to elastic waves traveling through the earth.

slowness

Inverse of velocity, in seconds/degree; a large slowness has a low velocity.

Solaris

Name of the operating system used on Sun Microsystems hardware.

SQL

Structured Query Language; a language for manipulating data in a relational database.

station

Collection of one or more monitoring instruments. Stations can have either one sensor location (for example, BGCA) or a spatially distributed array of sensors (for example, ASAR).

station processing

Processing based on data from a single station.

surface wave

Seismic wave propagating along the surface of the earth.

SWIM

Surface Wave Identification and Measurement.

T

time, epoch

See epoch time.

Tuxedo

Transactions for UNIX Extended for Distributed Operations.

tuxshell

Process in the Distributed Processing CSCI used to execute and manage applications. See IPC.

U

UNIX

Trade name of the operating system used by the Sun workstations.

W

waveform

Time-domain signal data from a sensor (the voltage output) where the voltage has been converted to a digital count (which is monotonic with the amplitude of the stimulus to which the sensor responds).

wfdisc

Waveform description record or table.

Index

W